AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for ~~providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having an object cache for storing objects~~synchronizing object caches, the method comprising:~~ the steps of:~~

changing a ~~an~~ first object in a~~the~~ first system, a second object in a second system, and a third object in a third system, wherein each object contains at least one unchanged attribute;

determining a first~~n~~ object change set, a second object change set, and a third object change set, -which represents the changes made to ~~the~~ the first object, the second object, and the third object, respectively, and wherein an object change set contains an attribute if and only if the attribute has been changed;~~ in the first system; and~~

sending the first object change set from the first system directly to the second system ~~to~~ which causes the second system to use a second merging unit to apply the first object change set to the corresponding object in the second system;~~'s cache so as to synchronize the second system with the first system, wherein control is returned to a client coupled to the first system whether or not the second system has completed processing the object change set, wherein the second system is registered in the first system prior to sending the object change information from the first system to the second system, and wherein the~~

~~registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in~~

sending the second object change set from the second system directly to the third system which causes the third system to use a third merging unit to apply the second object change set to the corresponding object in the third system; and

sending the third object change set from the third system directly to the first system which causes the first system to use a first merging unit to apply the third object change set to the corresponding object in the first system.

2. (Currently amended) The method as claimed in claim 1 further comprising a step of establishing a communication link between the first system and the second system wherein the ~~distributing~~ sending step ~~distributes~~ sends the first object change ~~information~~ set from the first system to the second system through the communication link.

3. (Original) The method as claimed in claim 2 wherein the establishing step establishes the communication link based on a publish/subscribe protocol.

4. (Canceled).

5. (Currently amended) The method as claimed in claim 1 further comprising a step of sending the first object change ~~information~~ set to a database for updating the corresponding object in the database ~~with the object change information~~.

6. (Currently amended) The method as claimed in claim 5 further comprising the steps of:

receiving an error message from the database ~~when~~ if the updating of the corresponding object in the database fails; and

discarding the first object change ~~information~~ set prior to the ~~distributing~~ sending step in response to the error message.

7. (Canceled).

8. (Canceled).

9. (Canceled).

10. (Canceled).

11. (Canceled).

12. (Currently amended) A method for ~~providing object change information from a first system to a second system for synchronizing the second system with the first system, the first system having a first object cache for storing one or more objects and the second system having a second object cache for storing one or more objects~~ synchronizing object caches, the method comprising: ~~the steps of:~~

receiving a first object change set at a first system, a second object change set at a second system, and a third object change set at a third system, which represent the changes made to a first object on the first system, a second object on the second system, and a third object on the third system, respectively;

wherein each object contains at least one unchanged attribute, and wherein an object change set contains an attribute if and only if the attribute has been changed;

sending the first object change set from the first system directly to the second system which causes the second system to use a second merging unit to apply the first object change set to the corresponding object in the second system;

sending the second object change set from the second system directly to the third system which causes the third system to use a third merging unit to apply the second object change set to the corresponding object in the third system; and

sending the third object change set from the third system directly to the first system which causes the first system to use a first merging unit to apply the third object change set to the corresponding object in the first system.~~determining object change sets which represent changes made to objects in the first system; and~~

~~sending the object change sets from the first system to the second system to cause the second system to apply the object change sets to corresponding objects in the second object cache so as to synchronize the objects in the second cache of the second system with the changed objects in the first system, wherein control is returned to a client coupled to the first system whether or not the second system has completed processing the object change set, wherein the second system is registered in the first system prior to sending the object change information from the first system to the second system, and where the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.~~

13. (Currently amended) The method as claimed in claim 12 further comprising a step of establishing a communication link between the first system and the second system wherein the ~~distributing~~ sending step ~~distributes~~ sends the first object change ~~information~~ set from the first system to the second system through the communication link.

14. (Canceled).

15. (Currently amended) The method as claimed in claim 12 further comprising a step of sending the first object change ~~information~~ set from the first system to a database for updating the corresponding object in the database ~~with the object change information~~.

16. (Currently amended) The method as claimed in claim 15 further comprising the steps of:

receiving an error message from the database ~~when~~ if the updating of the corresponding object in the database fails; and

discarding the first object change ~~information~~ set prior to the ~~distributing~~ sending step in response to the error message.

17. (Canceled).

18. (Canceled).

19. (Canceled).

20. (Canceled).

21. (Canceled).

22. (Canceled).

23. (Canceled).

24. (Canceled).

25. (Canceled).

26. (Canceled).

27. (Canceled).

28. (Canceled).

29. (Canceled).

30. (Canceled).

31. (Canceled).

32. (Currently amended) A ~~Computer~~ computer-readable storage-~~media~~ medium storing instructions ~~for use in the execution in a computer of~~that when executed by a computer cause the computer to perform a method for ~~providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having~~

~~an object cache for storing objects~~synchronizing object caches, the method comprising:~~ the steps of:~~

receiving a first object change set at a first system, a second object change set at a second system, and a third object change set at a third system, which represent the changes made to a first object on the first system, a second object on the second system, and a third object on the third system, respectively;

wherein each object contains at least one unchanged attribute, and wherein an object change set contains an attribute if and only if the attribute has been changed;

sending the first object change set from the first system directly to the second system which causes the second system to use a second merging unit to apply the first object change set to the corresponding object in the second system;

sending the second object change set from the second system directly to the third system which causes the third system to use a third merging unit to apply the second object change set to the corresponding object in the third system; and

sending the third object change set from the third system directly to the first system which causes the first system to use a first merging unit to apply the third object change set to the corresponding object in the first system.~~changing an object in the first system;~~

~~determining an object change set which represents changes made to the object in the first system;~~

~~sending the object change set from the first system to the second system to cause the second system to apply the object change set to the corresponding object in the second system's cache so as to synchronize the second system with the first system, wherein control is returned to a client coupled to the first system whether or not the second system has completed processing the object change set, wherein~~

~~the second system is registered in the first system prior to sending the object change information from the first system to the second system, and wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.~~

33. (Canceled).

34. (Canceled).